REMARKS

Claims 1-38 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 102, Anticipation

The Examiner has issued a final rejection of Claims 1-38, under 35 U.S.C. § 102(e), as being unpatentable over U.S. Patent No. 6,584,599 B1 to Huh et al. ("*Huh*"). This rejection is respectfully traversed.

With respect to Claims 1-38, the Examiner states:

> Claims 1-38 are rejected under 35 U.S.C. 102(e) as being unpatentable over Huh et al (US Patent no. 6,584,599 B1).
>
> The rejections are respectfully maintained and incorporated by references as set forth in the last office action.

Office Action dated November 24, 2003, page 2.

With respect to Claim 1 in the last Office Action, the Examiner states:

> As per claim 1, Huh et al teach a method for updating a current boot code in a data processing system in which the current boot code is used to load an operating system [fig. 3; col.1, lines 16-17; col. 2, lines 46-56; col. 3, lines 44-46; col. 4, lines 24-38], the method comprising the data processing system implemented steps of:
>
> loading a current boot code from a non-volatile memory [inherent step in process of booting a system; col. 3, lines 33-35; col. 4, lines 24-25];
>
> initiating a boot sequence using the current boot code [fig. 3; col. 4, lines 24-25];
>
> searching a storage device for an updated boot code [fig. 3; col. 4, lines 24-38; searching if new firmware is present or not and firmware is an ordered set of instructions and/or data that is used in booting computational system; col. 1, lines 16-17]; and
>
> updating the current boot code [fig. 3; col. 4, lines 24-38; processor determines whether any new firmware is present to upgrade or replace the old firmware].

Office Action dated July 28, 2003.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

As stated in the Applicants' response to the last Office Action, each and every element of the presently claimed invention is not shown (or suggested) within *Huh* arranged as in rejected Claim 1. Claim 1 reads as follows:

> 1. A method for updating a current boot code in a data processing system in which the current boot code is used to load an operating system, the method comprising the data processing system implemented steps of:
>
> loading a current boot code from a non-volatile memory;
>
> initiating a boot sequence using the current boot code;
>
> **searching a storage device for an updated boot** code for the operating system in response to initiating the boot sequence; and
>
> **updating the current boot code in the non-volatile memory** prior to loading the operating system for the data processing system if the updated boot code is present. (Emphasis added.)

Specifically, as stated in the Applicants' response to the last Office Action, *Huh* does not show, disclose (or suggest) an updated boot code, searching a storage device for an updated boot code, and updating the current boot code in the non-volatile memory, as in Claim 1. As such, in addition to the Applicants' arguments and comments given below, the arguments the Applicants made in response to the Office Action dated July 28, 2003 are incorporated herein by reference (See, Applicant's Response to Office Action dated October 28, 2003).

In this final Office Action, the Examiner states:

> In the remarks, applicants argued in substance that (1) Huh does not show or disclose searching a storage

> device for an updated boot code, and updating the current boot code in the non-volatile memory; (2) Huh teaches that the permanent boot code cannot be updated.
>
> As to point (1), Huh clearly discloses for looking whether any new firmware is present to upgrade or replace the old firmware [col. 4, lines 24-30] and also expressly teaches that the new firmware is typically stored on writeable, archival media [col. 3, lines 38-40. Plus, the new firmware can be loaded from a disk or uploaded or downloaded from another computational component [col. 3, lines 44-46]. Therefore, as shown in fig. 3, the decision box "IS NEW FIRMWARE PRESENT?" clearly indicates of a search process.
>
> As to point (2), Huh discloses that the boot code as whole comprises of one permanent initial boot initiating part [fig. 1, boot code 42] and second upgradeable or changeable part [fig. 1, firmware 54]. Firmware is an ordered set of instructions and/or data that is used in booting a computational system [col. 1, lines 16-17]. Huh expressly discloses that this permanent boot code directs the processor to read any new firmware as part of the boot sequence, which requires the processor to determine whether any new firmware is present to upgrade or replace the old firmware [col. 4, lines 24-30].

Office Action dated November 24, 2003, page 3. Notwithstanding the Examiner's statement, nowhere in these sections does *Huh* disclose, teach or suggest an updated boot code, searching a storage device for an updated boot code, and updating the current boot code, as recited in Claim 1.

Huh discloses a permanent boot code and firmware. *Huh* does not disclose, teach or suggest that its permanent boot code is firmware or vice versa (never mind an updated boot code). For example, the Examiner cites the following sentence in *Huh* for this teaching: "Firmware is an ordered set of instructions and/or data that is used in booting a computational system." The Examiner then relies on this sentence to infer that new firmware is updated boot code. However, *Huh* clearly separates and distinguishes firmware (new and old) from boot code. For example, *Huh* describes a configuration in which "the validating step can be preceded by the steps of reading a (boot) code discrete from the firmware (the [boot] code typically being recorded in permanent ROM) and determining if new firmware is present." (Col. 2, lines 36-39). *Huh* also describes an embodiment with "a computational system containing firmware... that includes: (a) means for reading a boot code (e.g., a code stored in ROM) of a computational component; [and] (b) means for determining, in response to reading of the boot code, if firmware is present in the computational component...". Clearly, in these sections, *Huh* separates and distinguishes boot code from firmware. *Huh* also discloses that old firmware is stored in permanent (non-writeable) ROM, and new firmware is stored in non-permanent ROM (the non-permanent ROM is writeable). (Col. 3, lines 33-38). Once again, *Huh* separates and distinguishes boot code from firmware.

Similarly, and most telling is Figure 3 and its supporting description in *Huh*, which describes reading the permanent boot code (step 204), and then the boot code directing the processor to read any new firmware and determine if new firmware is present to upgrade or replace the old firmware (step 208). Once again, *Huh* clearly separates and distinguishes boot code from (old and new) firmware. Even more telling is the sequence of steps 236 and 240 in Figure 3, which executes new firmware (step 236), and then determines whether or not the system is up and running (step 240). If the system is not up and running, even after executing the new firmware, the flow in Figure 3 proceeds back to step 200, which initiates the boot sequence (step 200), and then reads the permanent boot code (i.e., the same boot code as before, and not new boot code or updated boot code, as recited in Claim 1).

Also, from a logical standpoint, boot code does not necessarily have to be firmware. For example, an exemplary boot code could include a "jump" instruction that tells a processor where to go to find a real startup program (e.g., firmware in a BIOS startup sequence).

Therefore, for the above-described reasons (e.g., at least because *Huh* clearly separates and distinguishes boot code from both new and old firmware), and for the reasons given in the Applicants' response dated October 28, 2003, the Applicants respectfully assert that *Huh* does not disclose, show or suggest an updated boot code, searching a storage device for an updated boot code, and updating the current boot code, as recited in Claim 1.

Also, independent Claims 10, 15, 23, 32, 37 and 38 contain features similar to those of Claim 1 (e.g., an updated boot code, searching a storage device for an updated boot code, updating the current boot code, and the like) and are patentable over *Huh* for the same reasons. Also, the dependent Claims 2-9, 11-14, 16-22, 24-31 and 33-36 depending from these independent claims are patentable for the same reasons. Additionally, these claims include other combinations of features not taught by *Huh*.

The Applicants respectfully assert that the rejection of Claims 1-38, under 35 U.S.C. § 102(e), has been overcome. Therefore, the Applicants request that the Examiner reconsider and withdraw this rejection, and allow the pending claims.

II. Conclusion

It is respectfully urged that the subject application is patentable over *Huh* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: Jan 26, 2004

Respectfully submitted,

Richard J. Moura
Reg. No. 34,883
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants